ERICSSON

Audit and Security Logging

Cloud Execution Environment

INTERWORK DESCRIPTION

1/155 19-CRA 119 1864/5 Uen A

Contents

1 Scope

This document describes the Northbound Interface (NBI) of the Log Aggregator that is part of the Cloud Execution Environment (CEE).

The arrow between the Log Aggregator and the Log Collector in Figure 1 represents the Northbound Interface of the Log Aggregator.

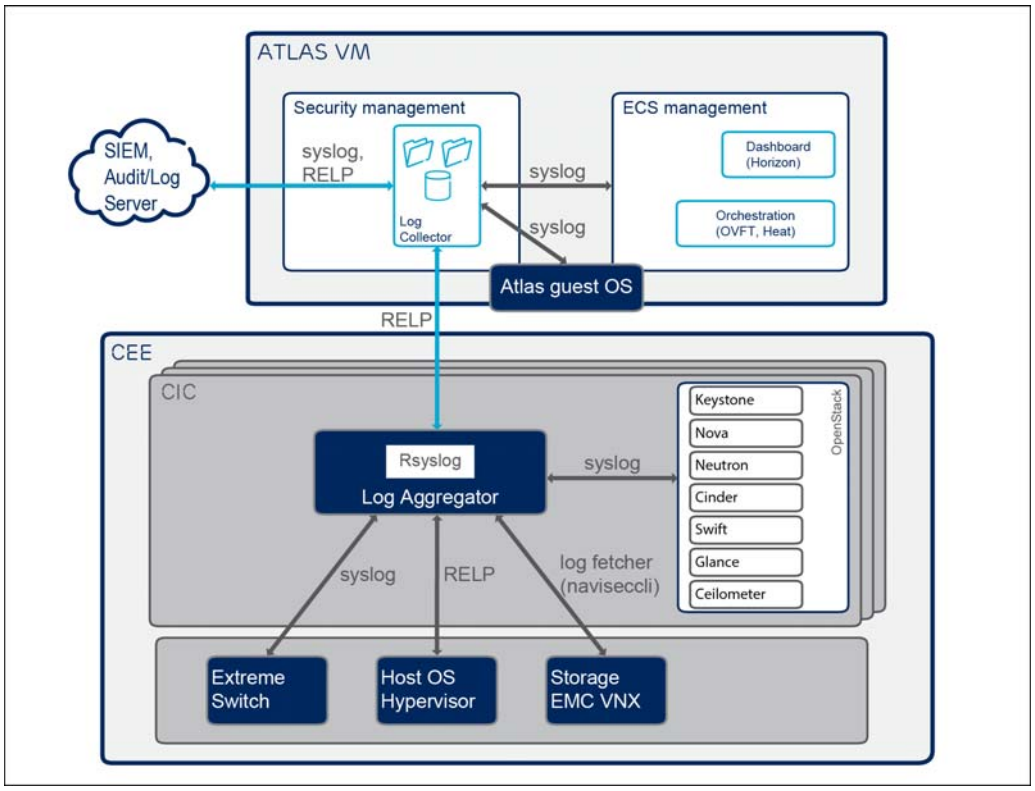

Figure 1 Security and Audit Logging

2 General

The Log Aggregator sends records over the Reliable Event Logging Protocol (RELP).

Event Sources

The audit and security logging system collects logs originating from the following event sources:

- Compute nodes
- Controller nodes
- Top-of-rack switches (Extreme)
- Storage back-end (EMC-VNX)

Audit and security log records from these nodes are either asynchronously pushed towards the Log Aggregator or periodically fetched in the case of EMC-VNX.

Log Aggregator

Security and Audit event records from the event sources are aggregated at the Log Aggregator and forwarded towards the Log Collector. The Log Aggregator functionality is performed by the rsyslog instance on the controllers, and it also acts as the system log. High availability of the Log Aggregator is ensured by the HA-proxy front-end that performs health check of the Log Aggregator interfaces and distributes the load among the controllers.

Log Collector

Atlas acts as the collector of all audit and security events in ECS. Rsyslog is configured to receive audit events over the RELP protocol and store them in a separate file system. The log collector listens on port `20514` for RELP connections. The incoming events are stored in a separate file system mounted at `/log-collector`. The size of the log-collector filesystem is 10% of the mounted data volume. The log file (`/log-collector/audit.log`) is truncated and rotated if it reaches the specified size limit. For an example of audit.log contents, see Example 1.

Note: Audit events received by the Log Collector can be forwarded to one or multiple external Security Information and Event Management (SIEM) systems for further analysis. Supported Protocols for event forwarding are RELP and syslog over TCP.

```
<14>1 2016-04-22T17:28:33.151518+05:30 localhost audispd - - -  node=atlas type=DAEMON_START ⇒
msg=audit(1461326313.146:1040): auditd start, ver=2.3.2 format=nolog kernel=3.13.0-74-generic ⇒
auid=4294967295 pid=2039 subj=unconfined  res=success
<14>1 2016-04-22T17:28:33.252499+05:30 localhost audispd - - -  node=atlas type=KERNEL ⇒
msg=audit(1461326168.677:1): initialized
<14>1 2016-04-22T17:28:33.252513+05:30 localhost audispd - - -  node=atlas type=AVC ⇒
msg=audit(1461326169.081:2): apparmor="STATUS" info="AppArmor sha1 policy hashing enabled" ⇒
pid=1 comm="swapper/0"
<14>1 2016-04-22T17:28:33.252523+05:30 localhost audispd - - -  node=atlas type=AVC ⇒
msg=audit(1461326173.687:3): apparmor="STATUS" operation="profile_load" profile="unconfined" ⇒
name="/sbin/dhclient" pid=497 comm="apparmor_parser"
...
```

Example 1 audit.log Typical Contents

3 Description

RELP is a networking protocol for computer data logging in computer networks. It is based on the ideas of the syslog protocol but extends it to provide reliable delivery of event messages. It is most often used in environments where message loss is not acceptable.

RELP uses a client-server model with (mostly) fixed roles. The initiating part of the connection is called the client, the listening part is called the server.

RELP uses the Transmission Control Protocol (TCP) for message transmission. This provides basic protection against message loss, but does not guarantee delivery under all circumstances. When a connection is aborted, it cannot be reliably detected if the last messages sent have actually reached their destination. Contrary to the syslog protocol, RELP works with a backchannel, over which information of messages processed by the receiver is conveyed back to the sender. This enables RELP to always know which messages have been properly received, even in the case of a connection abort.

For more information, seeReference [1] and Reference [2].

3.1 Protocol Versions

The RELP protocol version 1 is used.

The support for RELP in rsyslog is provided by the `librelp` library, version 1.2.9.

The rsyslog 8.16.0 version is used.

4 Message/Signal Definition

RELP employs a command-response model, that is, the client issues commands to which the server responds. Each command is assigned with a (relatively) unique, monotonically increasing ID, called the Transaction Number (TXNR). Each response must include that ID. A command and its response is called a RELP transaction.

For more information, see Reference [1] and Reference [2].

5 Configuration of Security Events Logging

Security log entries contain facility and priority levels. For details, see Table 1 and Table 2.

Table 1 Available Message Facilities

Value	Keyword	Description or type of event
0	kern	kernel messages
1	user	user-level messages
2	mail	mail system
3	daemon	system daemons
4	auth	security/authorization messages
5	syslog	messages generated internally by syslogd
6	lpr	line printer subsystem
7	news	network news subsystem
8	uucp	UUCP subsystem
9	-	clock daemon
10	authpriv	security/authorization messages
11	ftp	FTP daemon
12	-	NTP subsystem
13	-	log audit
14	-	log alert
15	cron	clock daemon
16	local0	local use 0 (local0)
17	local1	local use 1 (local1)
18	local2	local use 2 (local2)
19	local3	local use 3 (local3)
20	local4	local use 4 (local4)
21	local5	local use 5 (local5)
22	local6	local use 6 (local6)
23	local7	local use 7 (local7)

Table 2 Available Priority Levels

Value	Priority as per syslog definitions
0	`emerg`
1	`alert`
2	`crit`
3	`err`
4	`warning`
5	`notice`
6	`info`
7	`debug`

In Example 2 a security message is shown, where

6 is priority level `info`

10 is facility level `security/authorization messages`.

```
6,10,May 25 11:35:18,localhost,su[20501]:, Successful su for root by atlasadm.
```

Example 2 Security message contents

The default rules for rsyslog are configured in the `/etc/rsyslog.d/50-default.conf` file.

To create separate directories for each facility, the below lines have been commented in the `/etc/rsyslog.conf` file:

```
#$PrivDropToUser syslog

#$PrivDropToGroup syslog
```

Follow the below procedure to configure security and events logging:

1. Assign separate logging files for facility levels, based on priority.

   For example:

   - To log only error messages in a particular file, add the below line to `50-default.conf`:

     `authpriv.=err /var/log/authpriv/authpriv.err`

   - To log all messages of a particular priority, for example priority above `error` in a particular file (`err`, `warning`, `notice`, `info`, `debug`), add this line to `50-default.conf`:

     `authpriv.err /var/log/authpriv/xxxx`

- To log system daemon messages of all priorities, add this line to `50-default.conf`:

  `daemon.* /var/log/daemon.log`

2. Restart the rsyslog service using the following command:

   `atlasadm@atlas:~$ sudo service rsyslog restart`

The `auth` and `authpriv` messages are divided according to priority levels as follows:

- `/var/log/authpriv/authpriv.info`
  contains purely informational messages (usually does not require any handling)

- `/var/log/authpriv/authpriv.notice`
  contains on-error conditions that might require special handling

- `/var/log/authpriv/authpriv.err`
  contains errors other than hard device errors

6 Functions and Procedure Declaration

For more information, see Reference [1] and Reference [2].

7 Constants Declarations

N/A

Reference List

[1] *RELP – The Reliable Event Logging Protocol, Rainer Gerhards,* *http://www.rsyslog.com/doc/relp.html*

[2] *Reliable Event Logging Protocol, from Wikipedia the free encyclopedia,* *http://en.wikipedia.org/wiki/Reliable_Event_Logging_Protocol*